Outlier Detection and Anomaly Detection via CUSUM

Gerardo De La O

2022

Outlier Detection (Visually and Statistically)

This section explores the detection of outliers using visual methods as well as using the 'outliers' package. More specifically, we will explore the presence of crime rate outliers on a "per State" basis.

Link to data description:http://www.statsci.org/data/general/uscrime.html

Preparation

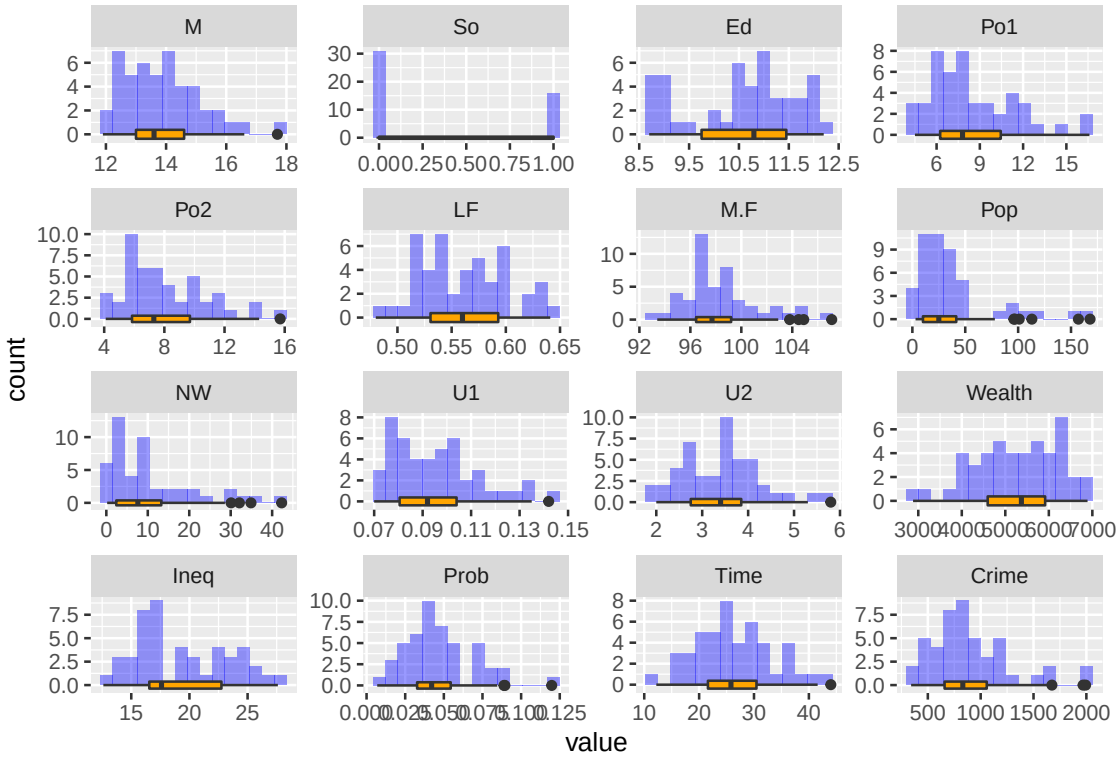
```
    library(caret)
    library(dplyr)
    library(ggplot2)
    library(tidyr)
    library(reshape2)
    library(outliers)
    library(lubridate)

    filename = 'http://www.statsci.org/data/general/uscrime.txt'
    data <- read.csv(filename, sep = '\t' ) #tab delimited
```

Detecting Outliers

The first thing to note is that outlier detection must be done before data normalization. If the reverse is true, then the data will not remain normal after outlier removal.

Visual Inspection One of the best ways to identify outliers is via Histograms and Boxplots. Here we plot the distribution of each variable with its corresponding box plot and immediately see that Crime has 5 potential outliers. However, it looks like a few of the other variables have a number of outliers themselves, so a reason could be that the high crime is correlated to the outliers in the other variables.

```
melted_data <- melt(data)

ggplot(data=melted_data, aes(x = value)) +
    geom_histogram(bins=15, fill='blue', alpha=0.4) +
    geom_boxplot(fill='orange') +
    facet_wrap(~variable, scales = "free")
```



Using Grubbs' Test Grubbs' Test is a statistical method used to determine, mathematically, whether a data point is an outlier. In R, the default implementation uses the formula $G = \frac{x - \hat{x}}{\sigma}$ where X is the value that we want to test. To run a Grubbs' test, our data should be approximately normally distributed, which is clear by looking at 'Crime' in the distribution plot above.

```
#Set up the test for observed potential outliers
test_vals = tail(sort(data$Crime),5)
test_vals
```

```
## [1] 1555 1635 1674 1969 1993
```

```
#Turns out we didn't need to test others! Placeholder for vectorized method here.
test = grubbs.test(data$Crime, type = 10, opposite = FALSE, two.sided = FALSE)
print(test)
```

```
##
##  Grubbs test for one outlier
##
## data:  data$Crime
## G = 2.81287, U = 0.82426, p-value = 0.07887
## alternative hypothesis: highest value 1993 is an outlier
```

Results

Initially, the goal was to test all 5 values shown as potential outliers (dots) in the boxplot for 'Crime'. However, the results of the test at our maximum value tells us that, while close, we are not able to reject the Null in favor of the alternative (p-value = .079)! Therefore, we conclude that our maximum is NOT an outlier. It follows that the rest of the values will also fail to reject the Null since they are smaller than the maximum. In practice, we would explore how the high values are affecting our model to get better context on how to handle them, but since we aren't building a model, we end this particular process here.

Real World Example - Anomaly Detection

I currently work at a B2B technology company specializing in conversational analytics. From a a business perspective, Change Detection can help our client companies gain insight into new or trending themes that occur in their phone conversations and chat interactions. As a matter of fact one of our Fortune 10 clients very recently asked if we could help them find which phrases "spike" during the period immediately after a new product launch. With this information, they could gain insight into what people are most excited about, what people are not liking, and in general what inquiries do people have about the new products.

If I were to apply CUSUM to solve this, I would use a combination of data and risk asessment to determine optimal values of C and T:

1. X values: The daily count of phrase occurrences. Consider only business hour volume (no nights and weekends)
2. Critical Value: I would start with 1 standard deviation as the starting parameter, then test against historical data. The historical results should give us a nice starting point, which we could then fine tune as we ingest live data.
3. Threshold: I would examine any flagged anomalies closely and determine if the change corresponds to something we want to measure. Then I would balance that with the perceived risk/cost implications. In this particular case, since false positives are easily dismissed without cost, we could start with a low threshold and gradually raise until it is tuned properly.

Anomaly Detection (CUSUM)

This sections explores the use of CUSUM to analyze temperature fluctuations in Atlanta, GA. We first examine a more graular approach, followed by a more systemic analysis of summer temperatures.

Unofficial "End of Summer" Identification

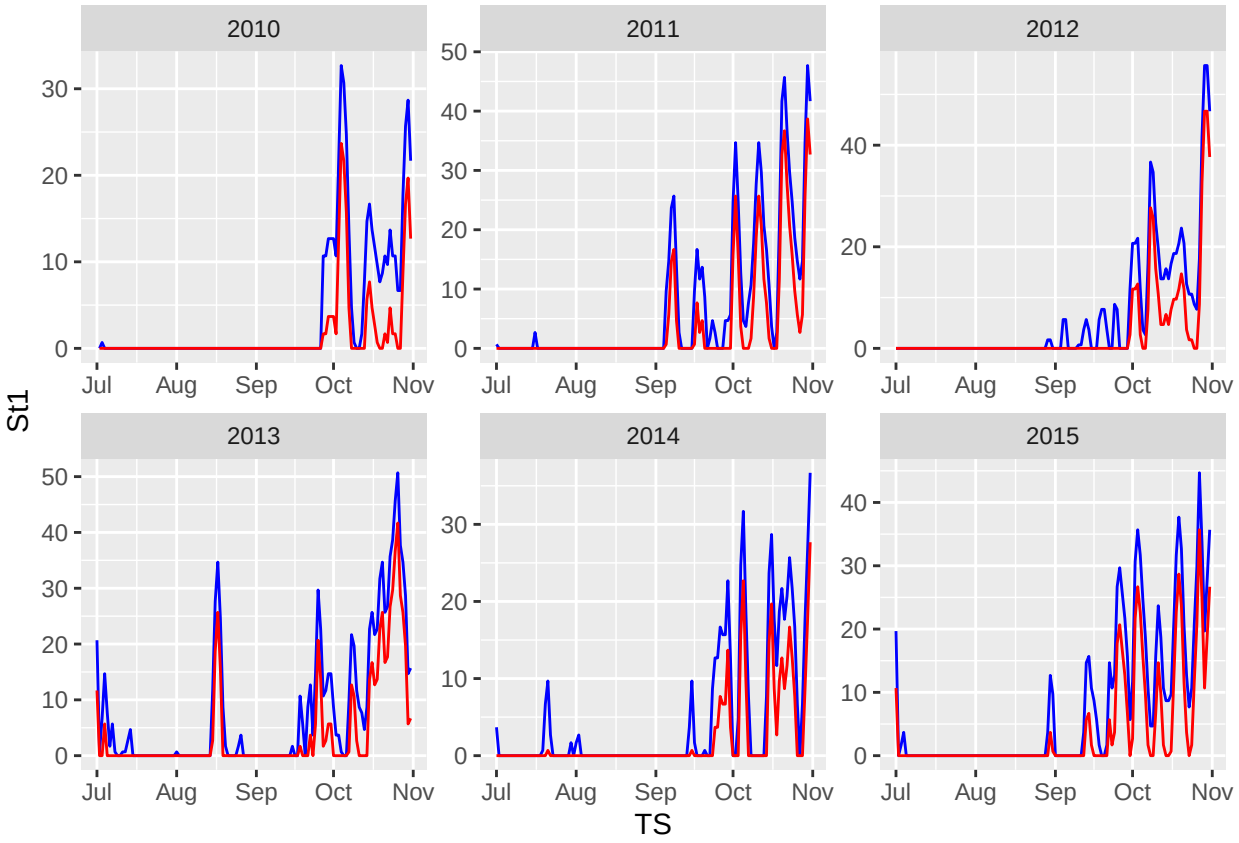
In this exercise, we use CUSUM to find the dates in which the temperatures transition from summer to fall. The first step will be to manipulate the data so we have the structure and requirements to implement the CUSUM algorithm.

```
filename = 'temps.txt'
data <- read.csv(paste0('data 6.2/',filename),sep='\t', check.names=FALSE)

keep <- c('DAY','2010','2011','2012','2013','2014','2015')
data = data[keep]

data_melted = melt(data, id.vars='DAY', variable.name='Year', value.name='Temperature')

mu = mean(data_melted$Temperature)
sd = sd(data_melted$Temperature)

C1 = .5
C2 = 5

data_melted['mu_X_C1'] =  mu - data_melted$Temperature - C1
data_melted['mu_X_C2'] =  mu - data_melted$Temperature - C2

df <- data_melted %>%
    unite(col='TS', c('DAY', 'Year'), sep='-') %>%
    mutate(S1 = mu_X_C1 + lag(mu_X_C1), S2 = mu_X_C2 + lag(mu_X_C2)) %>%
    rowwise() %>%
    mutate(St1 = max(0, S1), St2 = max(0, S2))
```

```
df$TS <- as.Date(df$TS, "%d-%b-%Y")

print(df, n=5)
```

```
## # A tibble: 738 x 8
## # Rowwise:
##   TS         Temperature mu_X_C1 mu_X_C2     S1     S2    St1   St2
##   <date>           <int>   <dbl>   <dbl>  <dbl>  <dbl>  <dbl> <dbl>
## 1 2010-07-01          87  -3.16    -7.66 NA      NA     NA        NA
## 2 2010-07-02          84  -0.159   -4.66 -3.32  -12.3   0         0
## 3 2010-07-03          83   0.841   -3.66  0.683  -8.32  0.683     0
## 4 2010-07-04          85  -1.16    -5.66 -0.317  -9.32  0         0
## 5 2010-07-05          88  -4.16    -8.66 -5.32  -14.3   0         0
## # ... with 733 more rows
```

```
ggplot(data = df, aes(TS)) +
    geom_line(mapping = aes(y = St1),color='blue') +
    geom_line(mapping = aes(y = St2), color='red') +
    facet_wrap(~year(TS), scales = "free")
```



End of Summer - Results

Based on the results of the CUSUM algorithm set to detect decreases, we can observe that the most consistent anomaly across the years tested occurs on the final days of September and usually extends into the beginning of October. Looking at the plots, the higher C value (red line - C=5) provides less noisy results,so we would keep that as our final model. Finally, for simplicity, let's just call September 30th the "Unofficial End of Summer"!

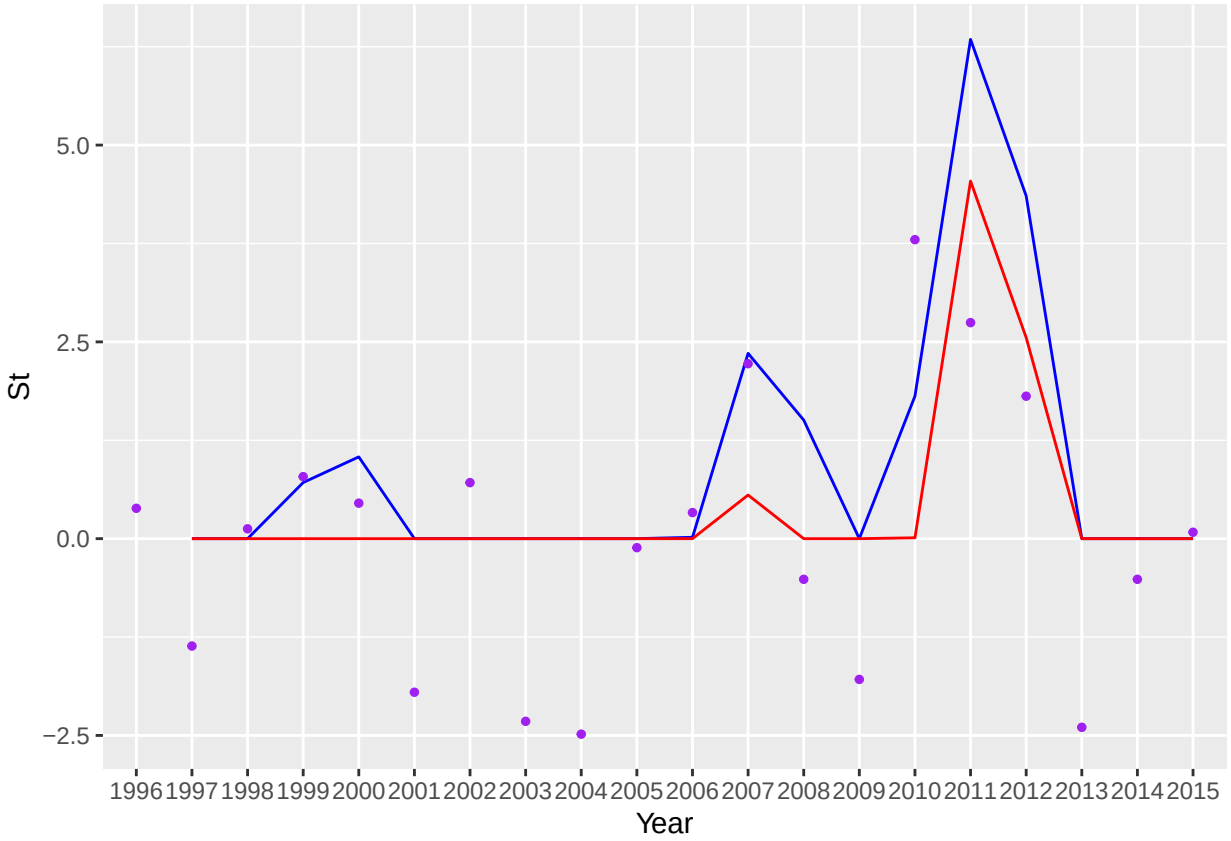
Climate Change - Analysis

This exercise is similar to the last one, but instead of looking at daily data points, we are looking at a much broader trend. In this exercise CUSUM will help us determine if summer temperatures have changed, and if so, when did the change occur. For simplicity, we will assume Fall begins on October 1st, which is consistent with our results from the previous question.

```
filename = 'temps.txt'
data <- read.csv(paste0('data 6.2/',filename), sep='\t', check.names=FALSE)


data <- data %>% filter(!grepl('Oct|Nov', DAY))

data_melted = melt(data, id.vars='DAY', variable.name='Year', value.name='Temperature')

data_m = data_melted %>% group_by(Year)  %>%
    summarise(AvgSummerTemp = mean(Temperature), .groups='drop')

mu = mean(data_melted$Temperature)
sd = sd(data_melted$Temperature)


C1 = .1
C2 = 1

data_m['X_mu_C1'] =  data_m$AvgSummerTemp - mu - C1
data_m['X_mu_C2'] =  data_m$AvgSummerTemp - mu - C2

df <- data_m %>%
    mutate(S1 = X_mu_C1 + lag(X_mu_C1), S2 = X_mu_C2 + lag(X_mu_C2)) %>%
    rowwise() %>%
    mutate(St1 = max(0, S1), St2 = max(0, S2))


print(df, n=5)
```

```
## # A tibble: 20 x 8
## # Rowwise:
##   Year  AvgSummerTemp X_mu_C1 X_mu_C2     S1     S2    St1   St2
##   <fct>         <dbl>   <dbl>   <dbl>  <dbl>  <dbl>  <dbl> <dbl>
## 1 1996           87.1   0.286  -0.614 NA     NA     NA        NA
## 2 1997           85.4  -1.46   -2.36  -1.18  -2.98   0         0
## 3 1998           86.8   0.025  -0.875 -1.44  -3.24   0         0
## 4 1999           87.5   0.688  -0.212  0.713 -1.09   0.713     0
## 5 2000           87.2   0.351  -0.549  1.04  -0.761  1.04      0
## # ... with 15 more rows
```

```
ggplot(data = df, aes(Year)) + ylab("St") +
    geom_line(aes(y=St1, group=1), color='blue') +
    geom_line(aes(y=St2, group=1), color='red') +
    geom_point(aes(y=AvgSummerTemp-mu, group=1), color='Purple', size=1)
```



Climate Change - Results

Based on the results of the CUSUM algorithm set to detect increases, we can observe that more anomalies have occurred in recent years, starting around 2006. From the plot, we can see there were 4 abnormally hot Summers in 2007 and 2010-2013, although the next 2 years had below average temperatures. Personally, I believe there could be some indicators that the summer climate is getting warmer (more anomalies in recent years), but based on this data alone there is not enough evidence to make that claim. Ideally, more data and additional models (other than CUSUM) would be able to provide more context to this hypothesis.